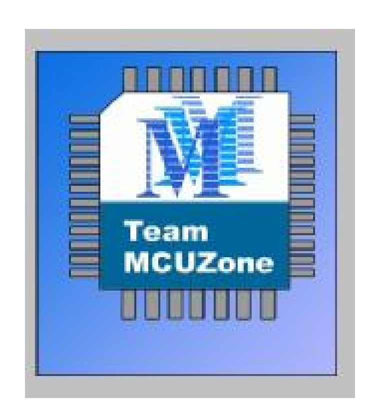
H-JTAG 使用说明



H-JTAG
twentyone
http://www.hjtag.com/



Team MCUzone
http://www.mcuzone.com

版本：Rev 1.0
更新日期：2006.10.28

更新记录

Rev 1.0:

文档创建

2006-10-28

目录

第 1 章 介绍

当前 ARM 的学习与开发非常流行，由于 ARM 的软件开发相对以前单片机而言更加复杂，硬件上的考虑也比较多，因此选择一个好的调试方法将可以使得开发的除错过程变得更加直接和简单。

现在市面上有很多可用于 ARM 调试的仿真器出售，然而其价格往往都比较贵。这些仿真器一般都有其专用的软件和硬件，在速度和 flash 编程等方面有各自的优势。然而对初学者而言，这些仿真器的成本都太高。而简易仿真器的出现，使得大家可以使用甚至自制 ARM 仿真器硬件。

有了调试器的硬件，还要加上调试代理软件，作为中介，将调试器前端软件（比如 AXD）的调试信息与目标板上的目标芯片交互，才能最终完成仿真的任务。目前，可以免费使用的简易 ARM 仿真器的代理软件很多，差别也比较大，主要表现在易用程度，目标器件支持，调试速度等方面。H-JTAG 作为近来新推出的简易 ARM 仿真器调试代理，其支持器件比较多，支持的调试器前端软件也比较多，特别是支持 keil，其调试速度也很有优势。

1.1 H-JTAG 介绍

H-JTAG 是由 twentyone 推出的一款免费调试代理软件。官方主页为：

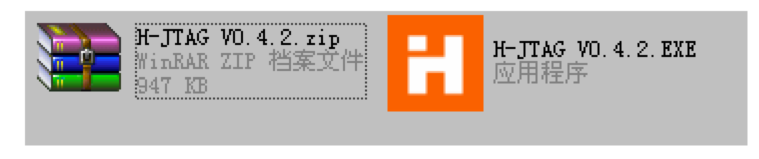
http://www.hjtag.com/

目前的版本为 0.4.2（2006 年 12 月 05 日），支持下列特性：

1. 支持 RDI 1.5.0 与 1.5.1;
2. 支持 ARM7 与 ARM9（包括 ARM9E-S 与 ARM9EJ-S）;
3. 支持 thumb 与 arm 指令集;
4. 支持 little-endian 与 big-endian;
5. 支持 semihosting;
6. 支持 wiggler, sdt-jtag 以及用户自定义的简易调试器硬件接口;
7. 支持 WINDOWS 9.X/NT/2000/XP;
8. 支持 flash 器件的编程

1.2 H-JTAG 安装

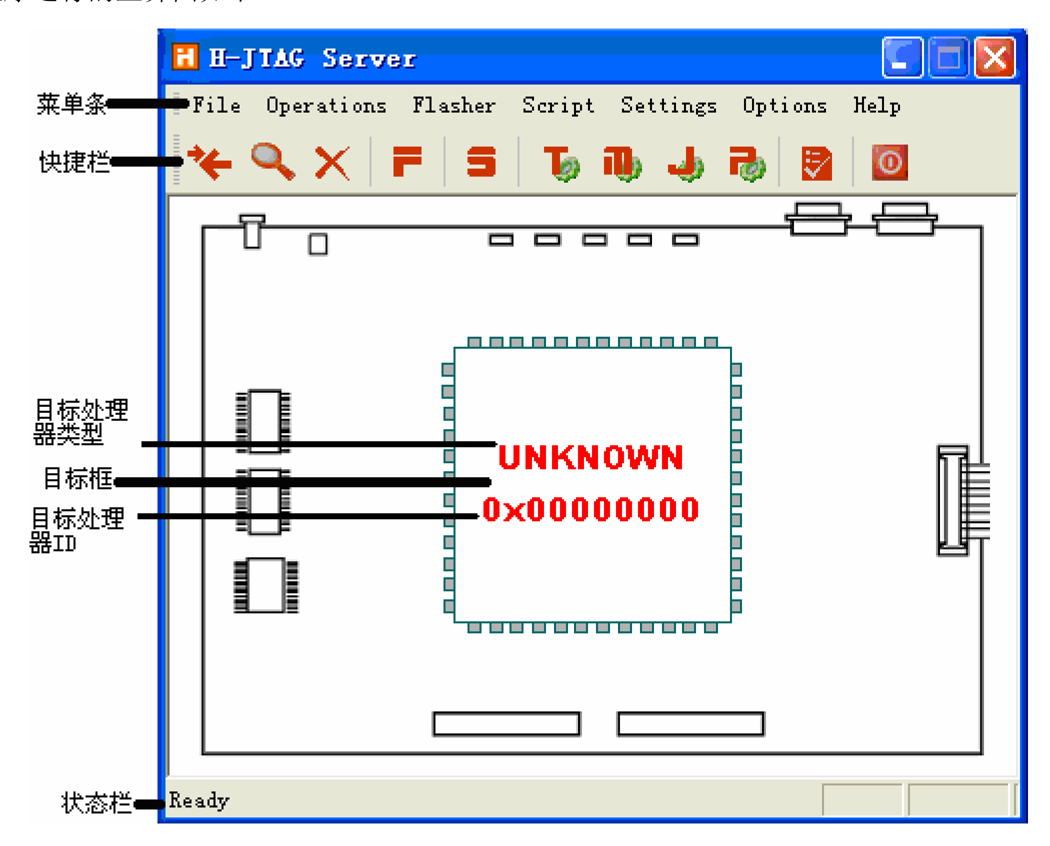
根据上面给出的链接，通过 download 页面下载 H-JTAG 压缩文件包，展开即可获得 H-JTAG 的安装文件，如下图：



双击右侧安装程序即可开始安装。安装完成，点击 H-JTAG.exe 即可开始运行 H-JTAG。

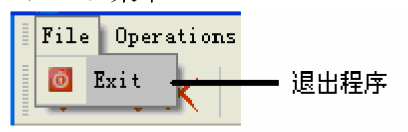


程序运行的主界面如下：



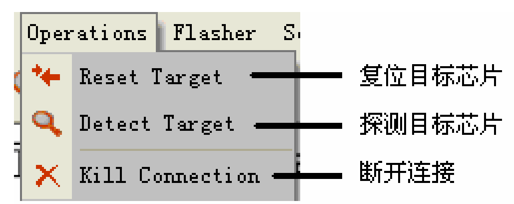
上图是未找到目标器件的画面。快捷栏中的工具图标与菜单条中具有相同图标的菜单功能一致，可以对照下面的菜单详细介绍：

1）File 菜单



退出程序：停止 H-JTAG 的运行。点击右上角的关闭按钮只会使 H-JTAG 最小化到系统托盘，而不会停止其运行。

2）Operations 菜单



复位目标芯片：可以使得目标器件复位

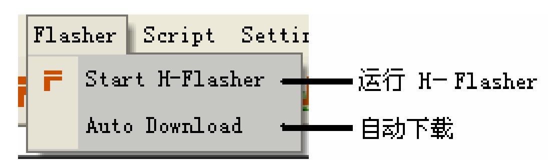
探测目标芯片：可以使得 H-JTAG 开始探测目标芯片；默认情况下，H-JTAG 在开始运行的时候会自动探测目标器件。如果探测不到目标芯片，会弹出下列对话框：



此时请检查硬件连接。

断开连接：在有调试前端软件(axd 等)连接的情况下，可以断开与调试器前端的连接。

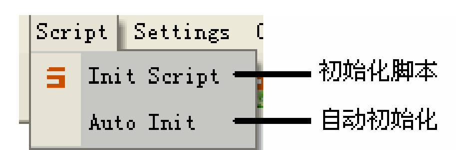
3) Flasher 菜单



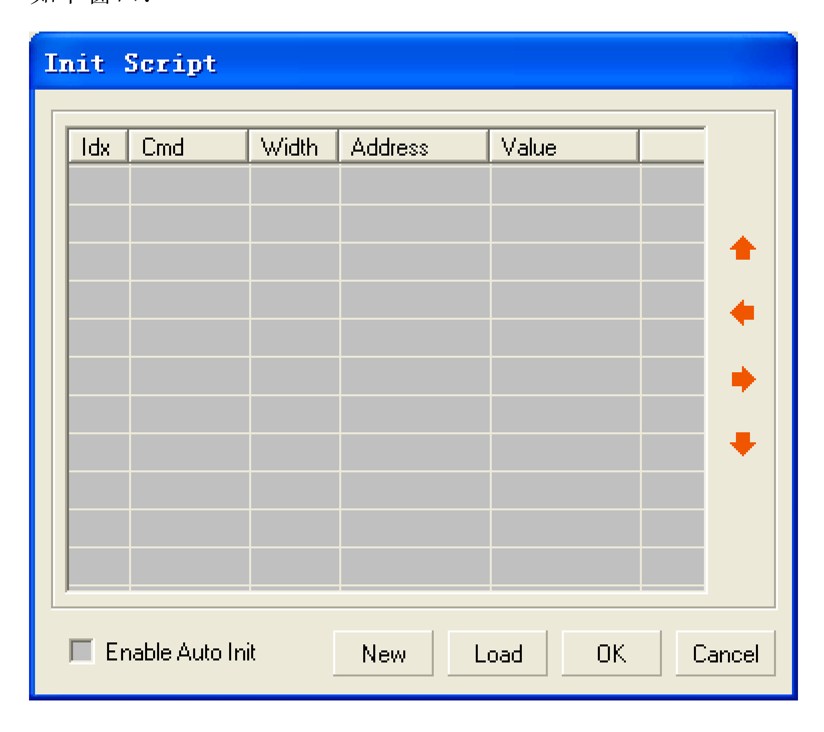
运行 H-Flasher：运行 H-Flasher 软件。该软件的具体功能将在第 3 章介绍。

自动下载：选中该功能后，调试前端软件在需要的时候可以自动下载程序到 flash。适用于在 flash 中调试软件。这样在调试的时候就会自动下载代码到 flash，可以省掉一步手动下载的过程。

4）Script 菜单



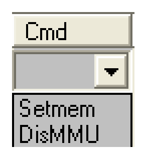
初始化脚本：该功能类似 axd 提供的初始化脚本功能。可以在调试器开始运行的时候自动完成一些对器件的初始化操作。比如禁止 watchdog，初始化 SDRAM 等。点击该功能将弹出如下窗口：



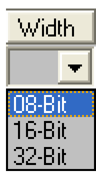
点击 将会添加一条新的条目：



点击 cmd 将会添加一个具体的命令：



目前支持的只有设置内存(Setmem)和禁止 MMU(DisMMU)。
Width 窗口可以选择所访问内存的宽度：



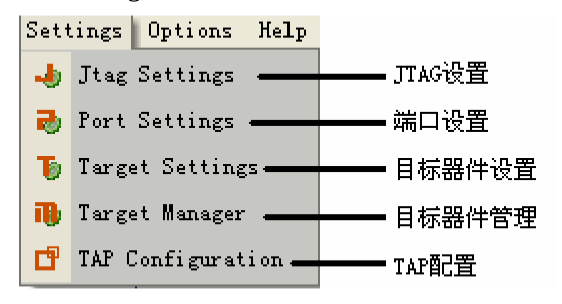
应该按照实际内存访问的宽度来访问。
Address 可以设置欲进行内存的操作的地址。
Value 就是往内存地址写的值。
填写完成的一个条目如下：



点击 可以再添加一条，点击 可以删除一个条目，点击 与 可以调整条目间的顺序。
下方的菜单与脚本的保存和运行有关：

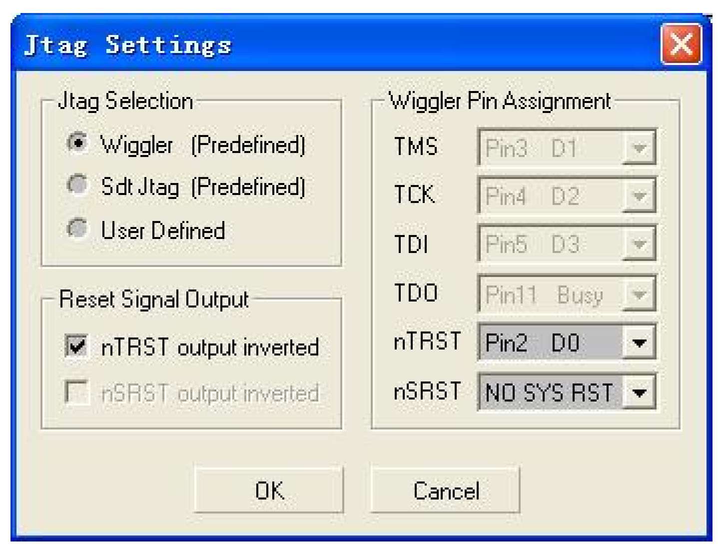


勾选了 Enable Auto Init，将会使得 H-JTAG 在运行时自动完成初始化操作。
点击 New 将清除所有的条目，Load 可以载入一个已经定义好的初始化文件。点击 OK 即可保存但前的设置到一个初始化文件，方便下次加载。点击 Cancel 将不保存改变。
自动初始化：勾选了该功能等同于勾选了初始化脚本中的 Enable Auto Init。
5）Settings 菜单



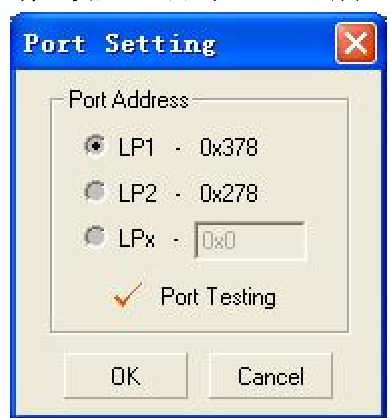
JTAG 设置：此菜单可以对 JTAG 的硬件进行设置，包括所使用简易仿真器的类型，目前支持 wiggler，sdt 与自定义的 JTAG 连接。如果选择 JTAG 的硬件为自定义，则需要在右侧选择 JTAG 信号线对应的并口线。注意两个复位信号线 nTRST 与 nSRST 是一直可以自定义的。必须根据硬件选择。Reset Signal Output 里可以选择两个复位信号的极性。下图没有选择

nSRST 信号，因此左侧复位信号取反中 nSRST 是禁止的。



Wiggler 与 SDT JTAG 的硬件电路图可以参考附录 A 与附录 B。

端口设置：可以设置 PC 的并口，如下：

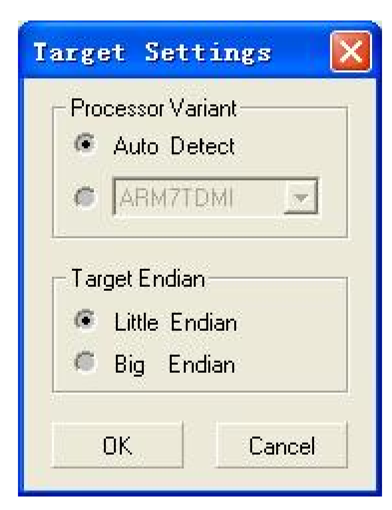


应根据 PC 的并口设置来选择，选择完成后可以运行 Port Testing，如果弹出如下错误：



则应该检查端口设置。

目标器件设置：此菜单可以选择目标器件的类型与 endian 的类型。处理器类型一般选择自动探测即可，如果实在探测不出，也可以试试手动指定。目标器件的对齐模式可以选择小端对齐或者大端对齐。选择完成点击 OK 之后，H-JTAG 即会开始探测目标器件。

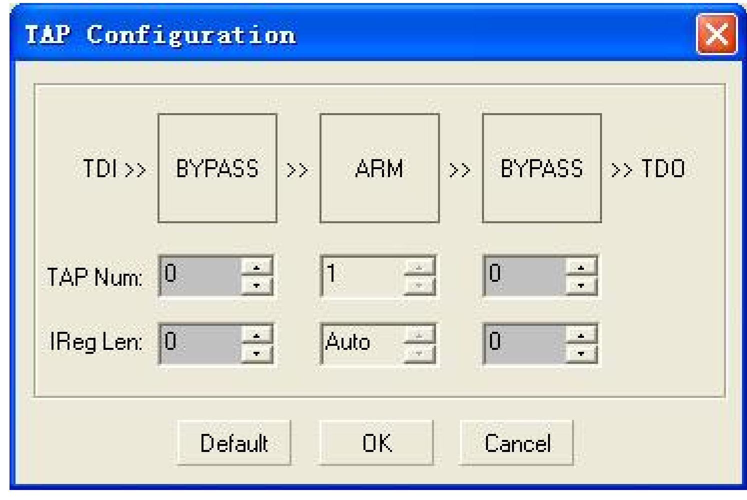


目标器件管理: 此菜单可以管理目标器件。

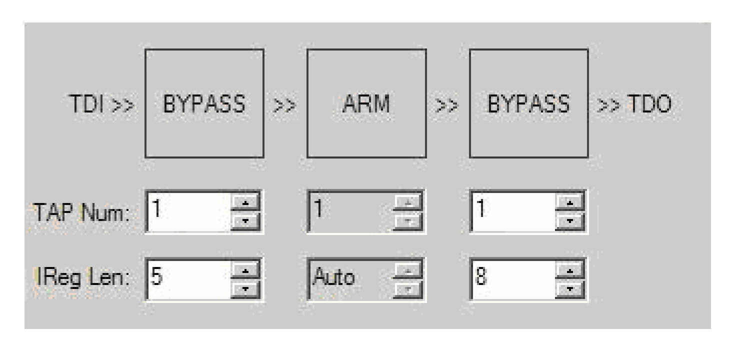


分为添加新的 ID 号以及删除已存在的 ID 号。如果目标器件的 ID 不在目前的处理器列表里，H-JTAG 将不能识别其处理器类型。使用该功能即可添加其 ID，并为其指定一个处理器类型，那么下次 H-JTAG 再次探测到该 ID 时，就会指出其处理器类型并进行相应的处理。删除与此过程相反。

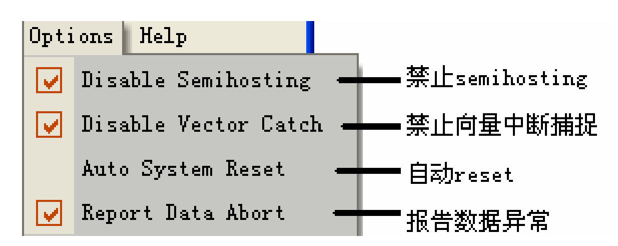
TAP 配置： 此功能用来配置目标的 TAP。



一般不用设置，但对于 TAP 特殊的目标器件必须手动设置。比如 STR912 就必须按照下图设置，才能正确使用：

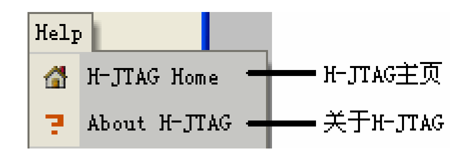


6）Options 菜单



禁止 Semihosting：禁止使用 semihostiong 功能。
禁止向量中断捕捉：一些调试软件会为异常向量(swi，data abort 等)保留断点，这会占用系统提供的断点资源。而 ARM7 系列只能提供两个硬件断点。如果勾选这个功能，在 flash 中调试的时候将不会有余下的断点资源用于单步等。因此一般应该勾选该选项。
自动 reset：勾选此功能将使得调试软件在必要时可以直接复位目标芯片。
报告数据异常：勾选此功能后，当处理器在调试过程中产生数据异常，H-JTAG 将会报告异常。

7）Help 菜单



H-JTAG 主页：点击后即可打开 H-JTAG 主页。
关于 H-JTAG：将会显示版本信息及作者的联系方式，如下图：



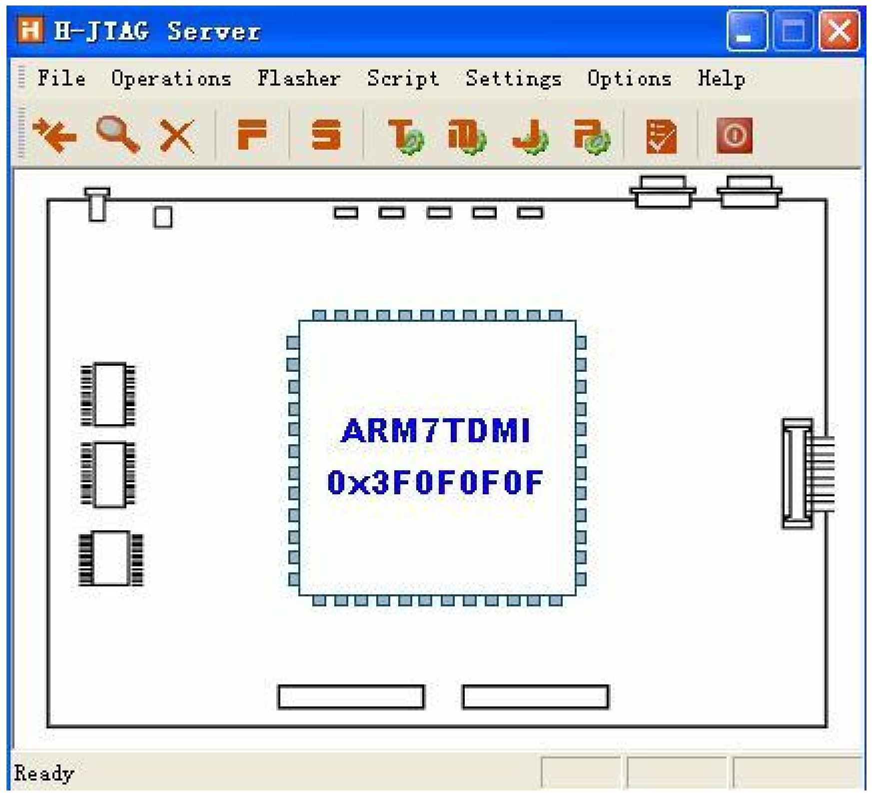
通过里面的链接可以方便的连接到 H-JTAG 主页，以及向作者反馈信息。

第 2 章 调试

H-JTAG 的主要功能是作为调试代理，使得调试器前端软件可以通过其与简易的 JTAG 通信，完成目标调试的任务。
由于支持 RDI 接口，H-JTAG 支持很多市面上流行的调试软件。下面将分别介绍在流行 IDE 下如何使用 H-JTAG 进行调试，硬件平台使用 mcuzone 的 S64-DEK 2.0 和 wiggler。

2.1 在 ADS1.2 中使用 H-JTAG 调试

使用 wiggler 连接 PC 和目标板，连接好开发板的 USB 线到 PC，运行 H-JTAG，如果连接正确，H-JTAG 将发现 ARM7TDMI 核心(上面步骤以下简称为：正确连接硬件)，如下图：

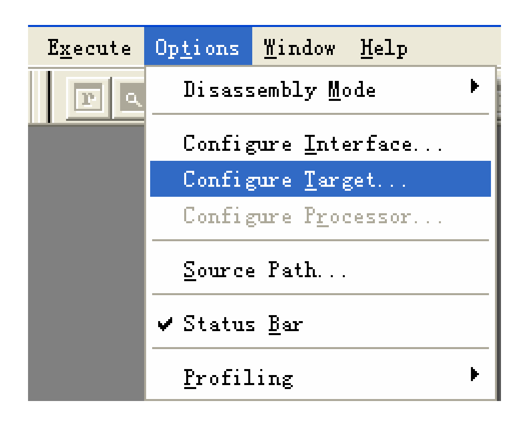


如果在器件类型中显示为 UNKNOW，或者出现如下对话框：

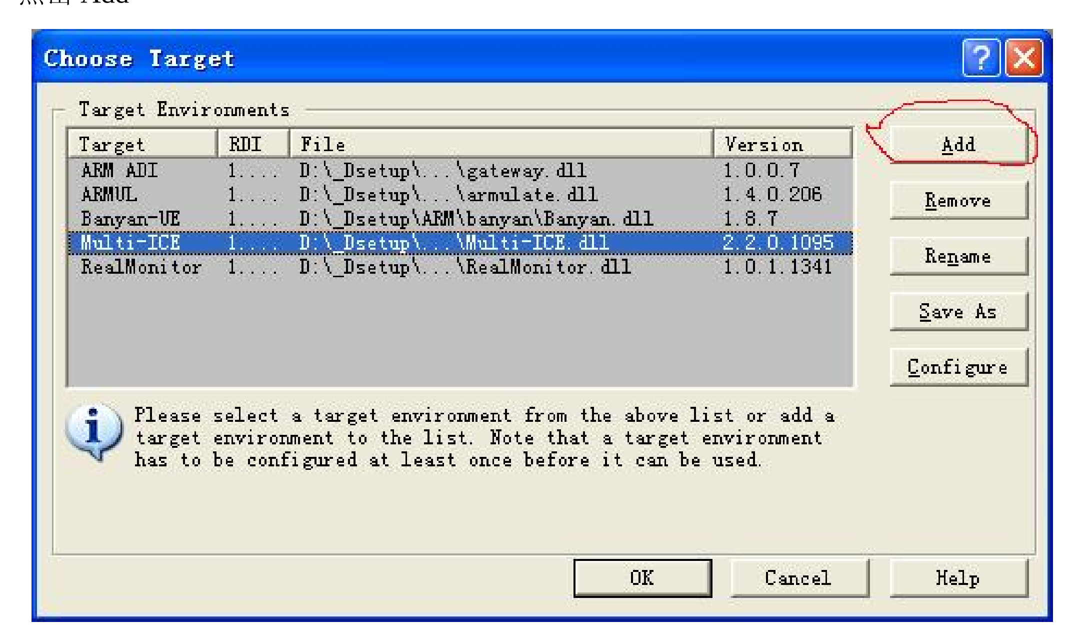


即说明 H-JTAG 未能与目标器件建立连接，应该检测硬件连接，直到出现正确的连接画面。

运行 AXD,首先要添加 H-JTAG 为新的 target。点击 Options à Config Target, 如下图



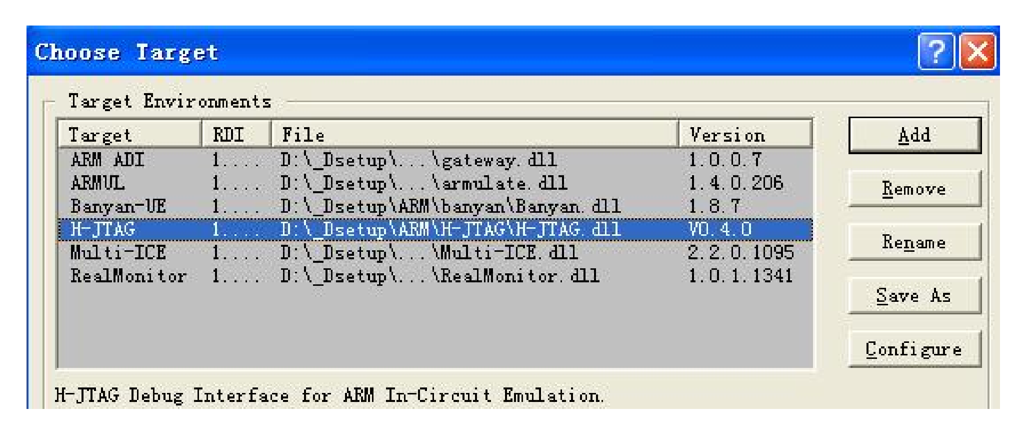
点击 Add



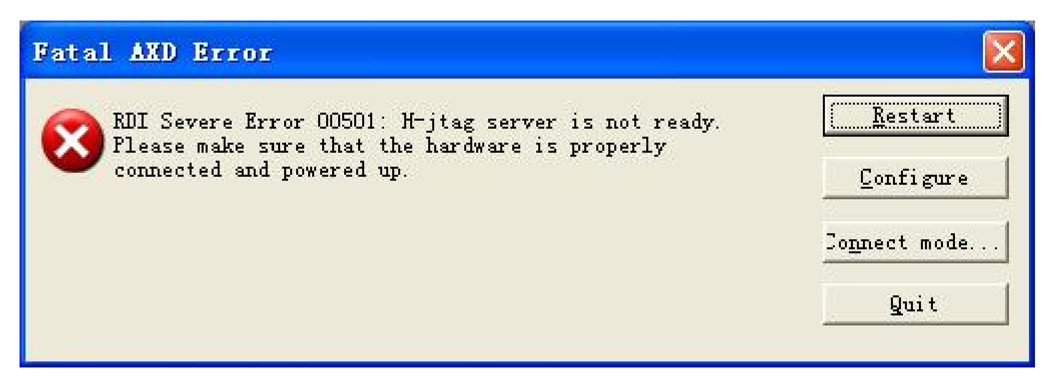
添加 H-JTAG 安装目录下的 H-JTAG.dll，



点击 “打开”，H-JTAG 即会被添加为一个新的 target，效果如图：

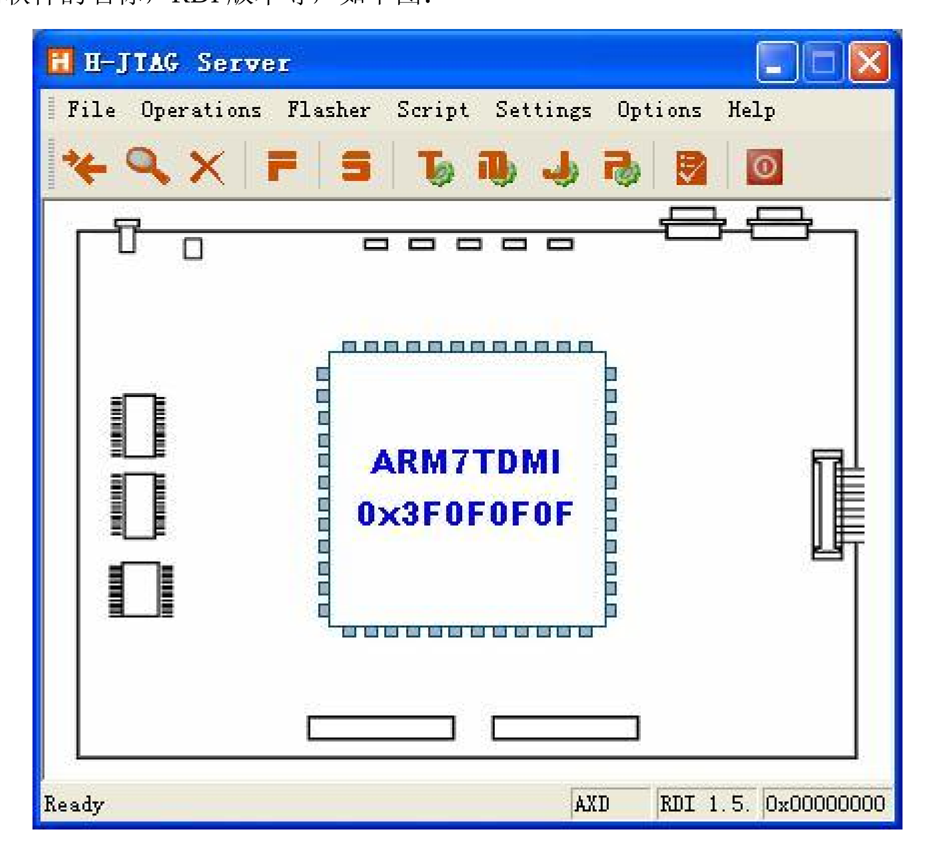


点击 OK 即可使用 H-JTAG 作为调试代理。如果出现如下对话框：



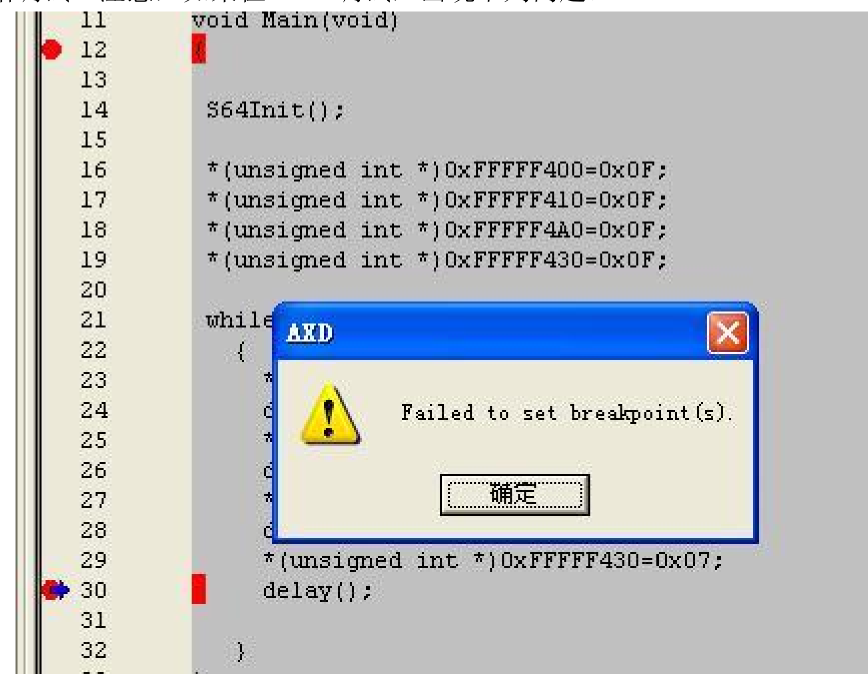
说明连接有问题。

当 AXD 连接上 H-JTAG 之后，H-JTAG 下方的状态栏会有详细的显示，包括调试器前端软件的名称，RDI 版本等，如下图：



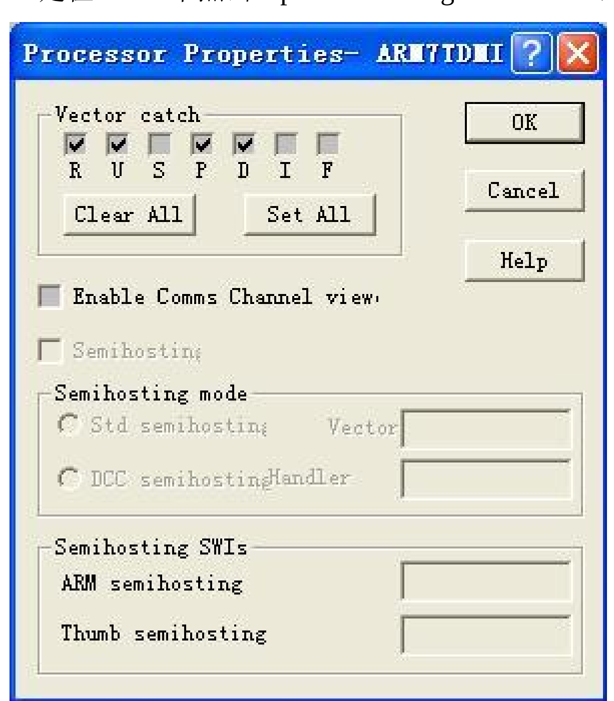
此时 AXD 也处于就绪状态，通过点击 File à Load Image…载入欲调试的 axf 文件即可

开始调试。注意，如果在 flash 调试，出现下列问题：

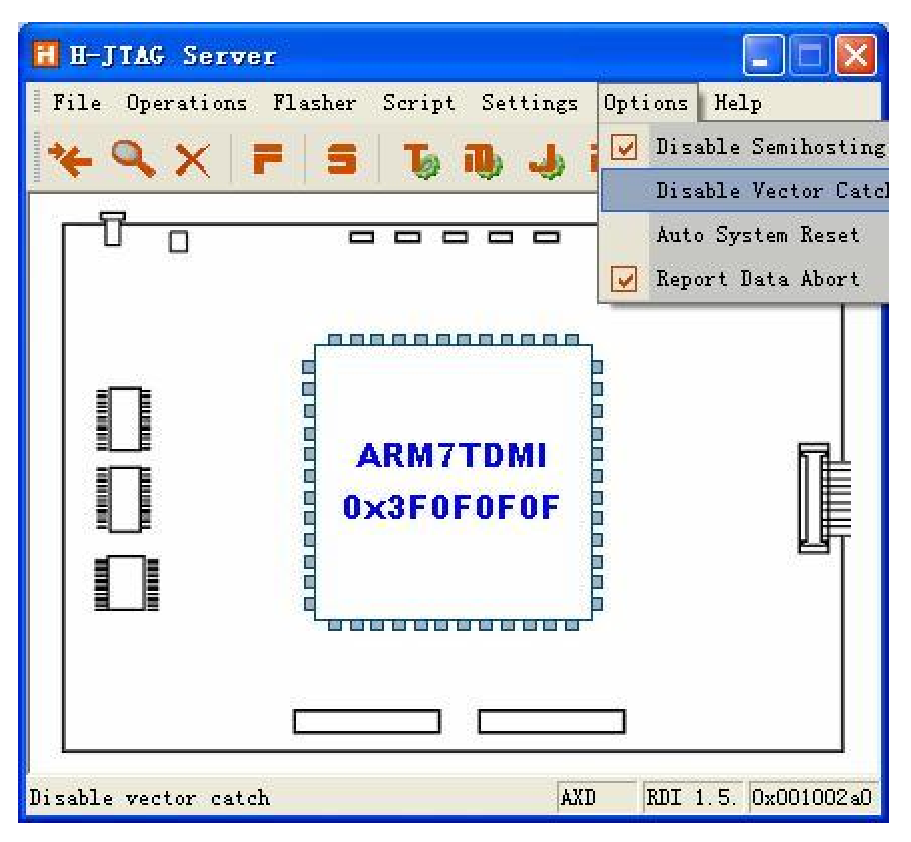


则说明断点设置过多，原因在于 ARM7 系列只有两个硬件断点，而在 flash 中调试时只能使用硬件断点，因此断点数量有限制。一般而言，系统将会保留一个硬件断点供单步调试时使用，因此实际所能使用的断点仅剩一个。需要注意的是，默认情况下，AXD 会为异常向量保留断点，这将造成在实际调试的时候没有任何断点可用，解决方法有两个：

一是在 AXD 中点击 Options à Config Processor…，点击 Clear All 去掉所有。



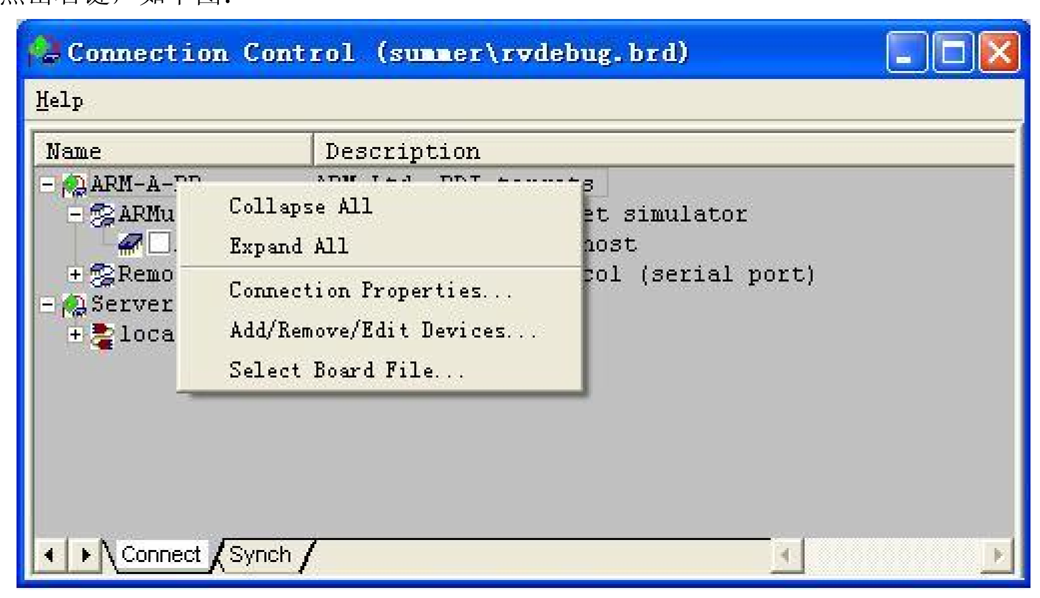
二是在 H-JATG 中直接禁止所有的向量捕捉，如下图：



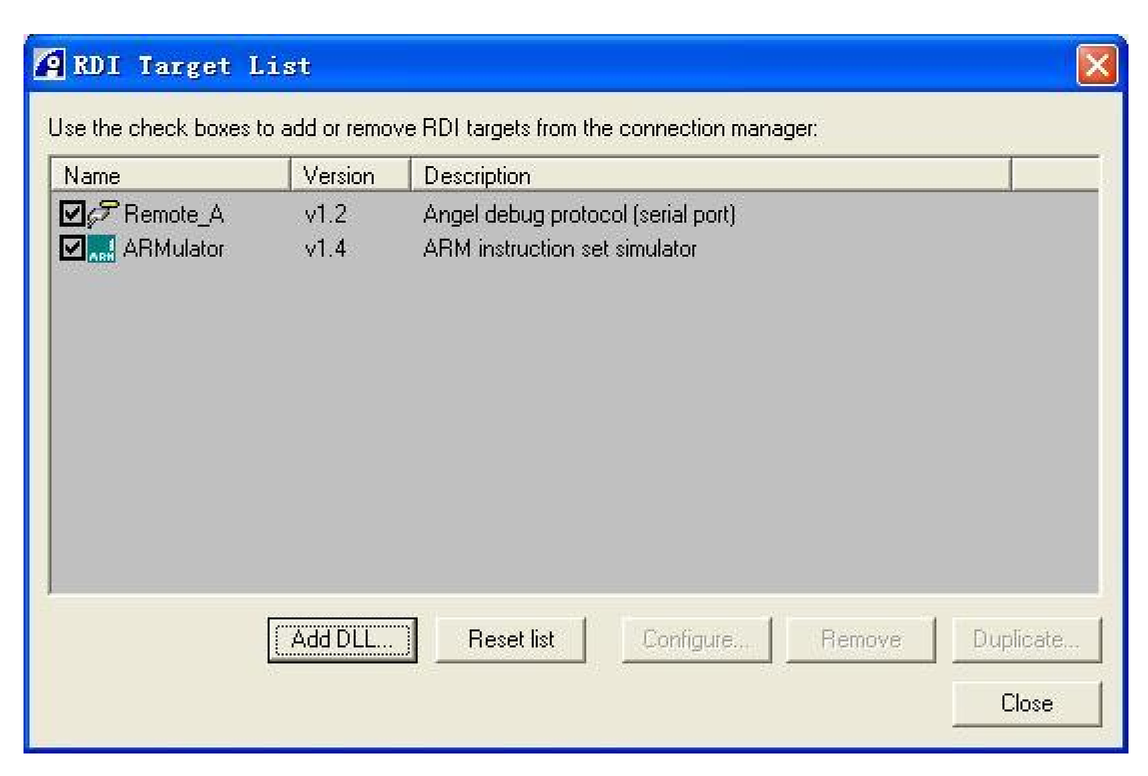
勾选 Disable Vector Catch 即可。

2.2 在 Realview2.2 中使用 H-JTAG 调试

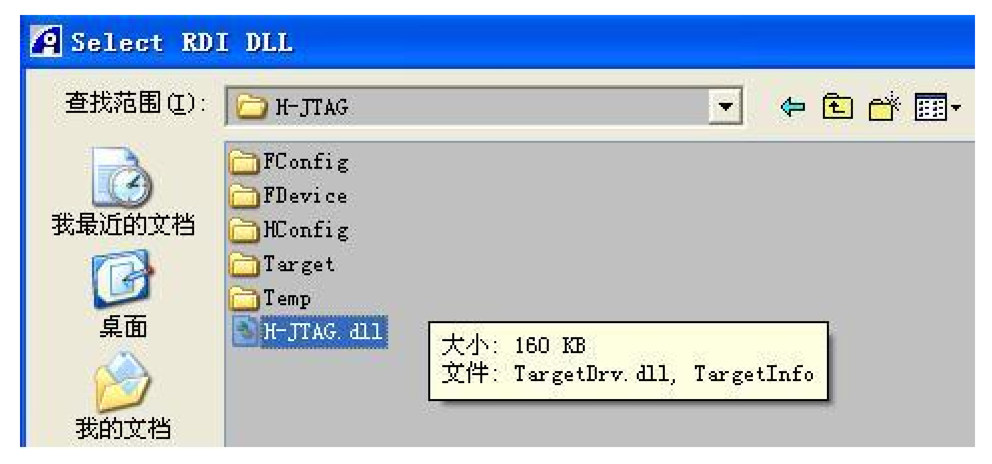
正确连接硬件后，运行 RVD，点击 Target à Connect to Target…,在弹出的连接控制窗口中点击右键，如下图：



点击 Add/Remove/Edit Devices…即会弹出下列窗口：



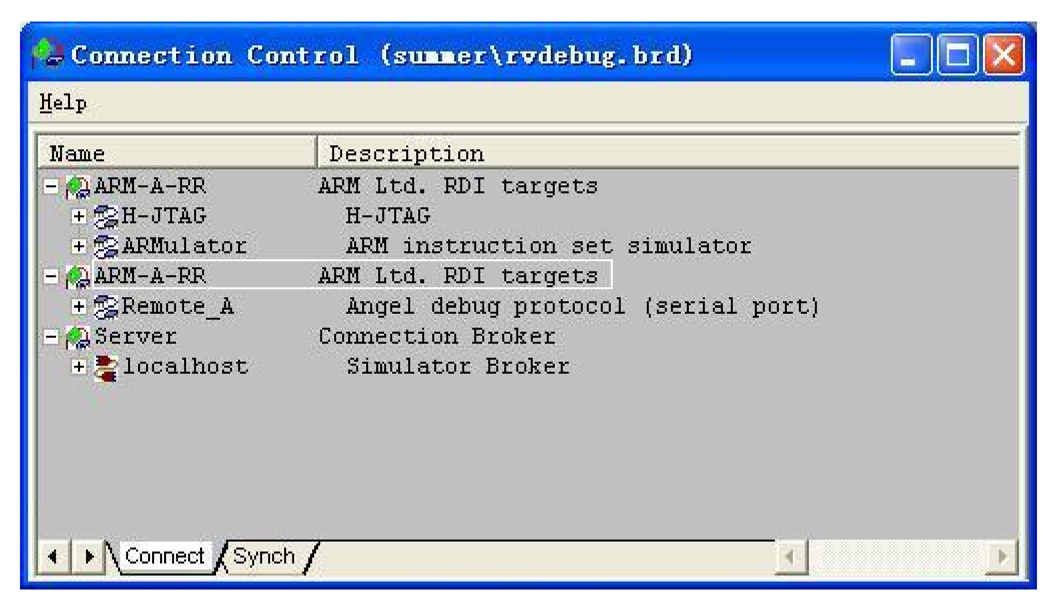
选择 Add Dll…，选择添加 H-JTAG.dll,



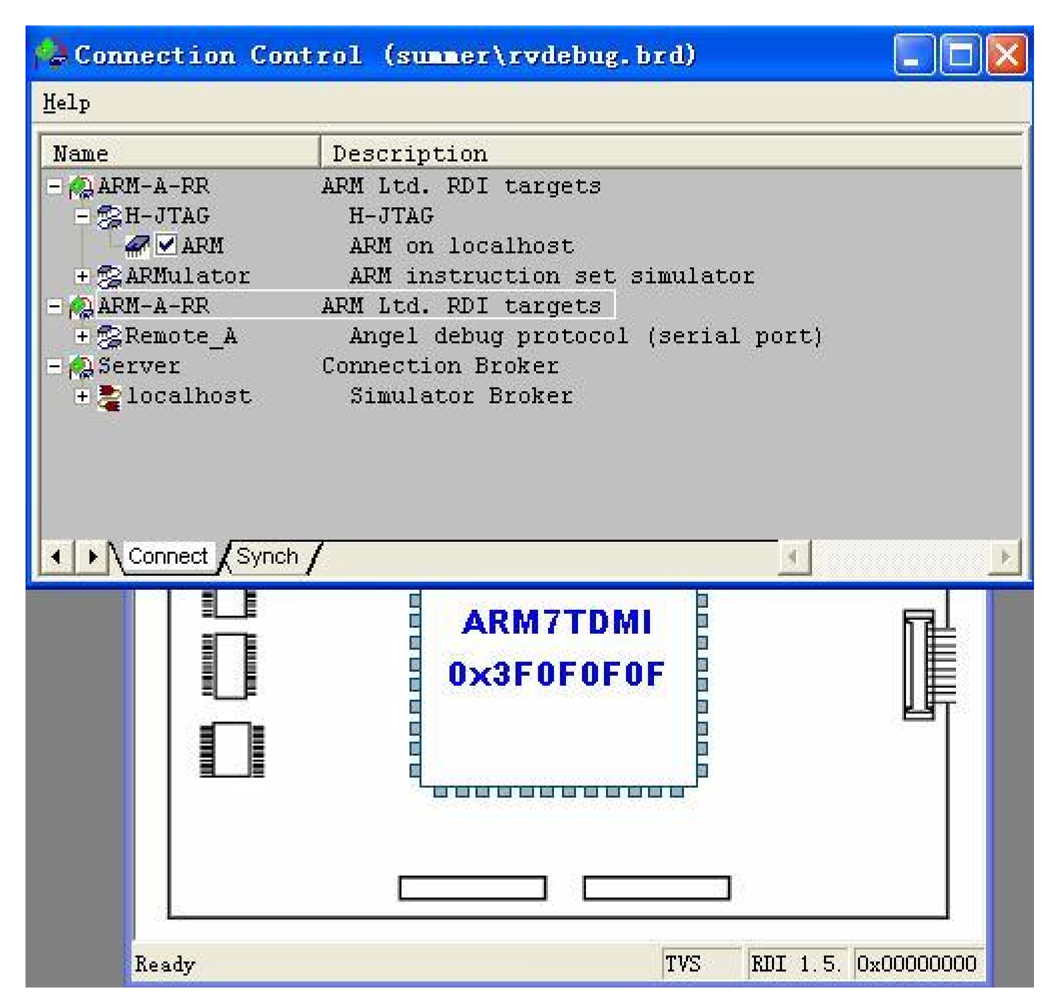
为 H-JTAG 设定名字即描述，比如直接叫做 H-JTAG。



点击 OK 之后，新的 RDI 目标即被添加到 RVD 中，如下图：

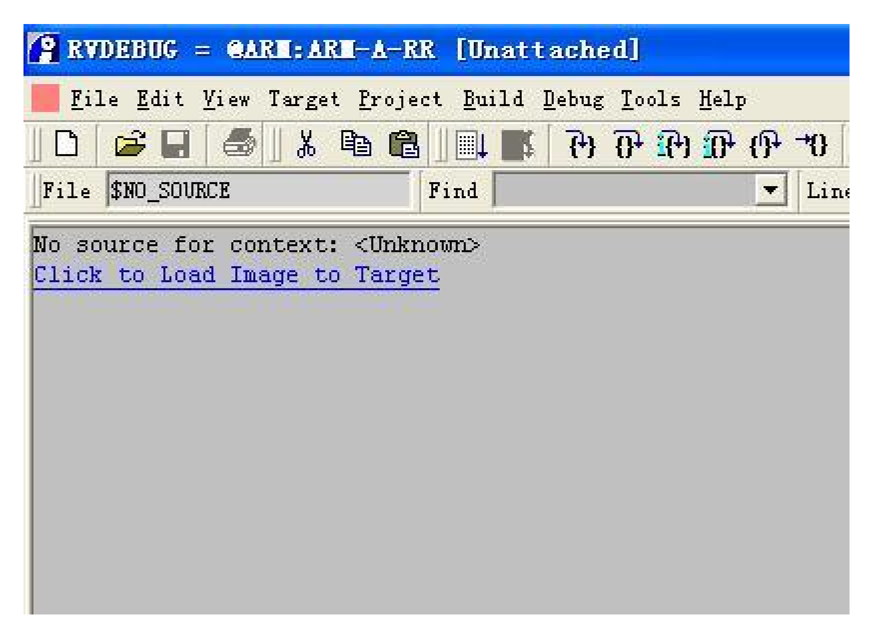


点开 H-JTAG 左侧的“＋”号，勾选下方的 ARM，即可与 H-JTAG 建立连接。

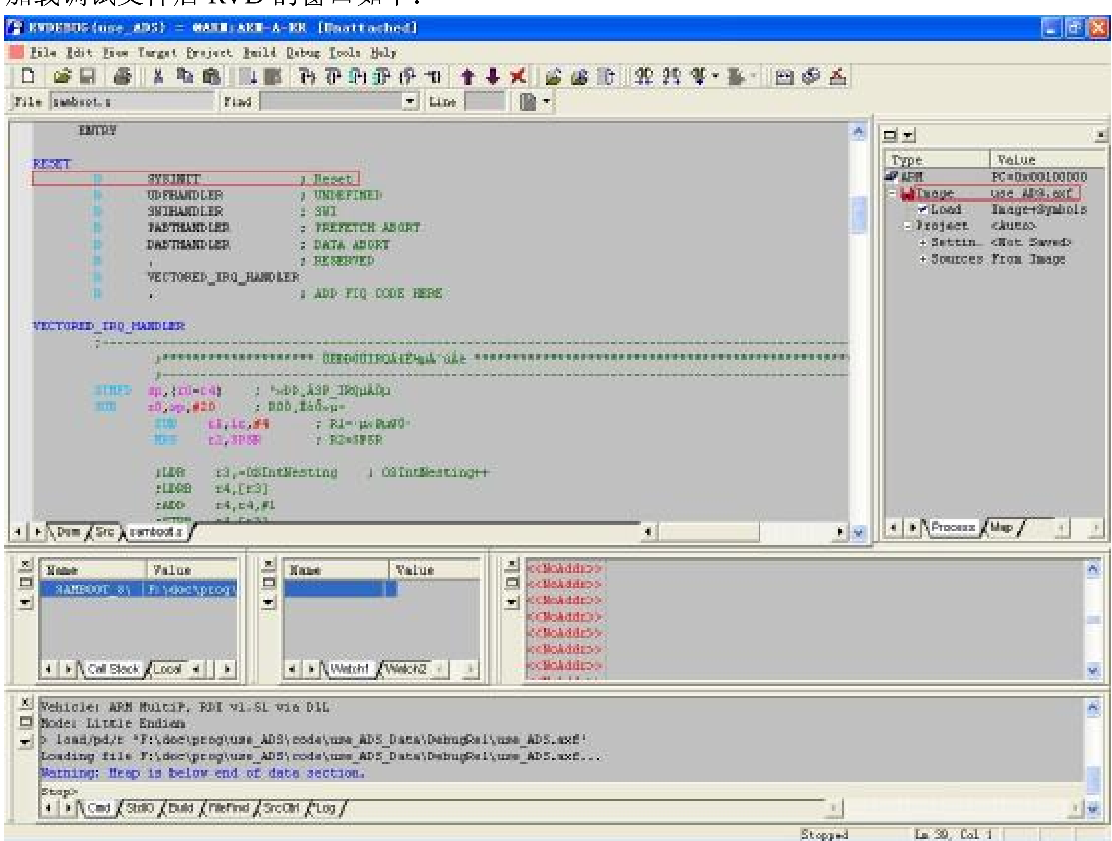


连接完成后的 H-JTAG 的状态栏将显示连接状态。

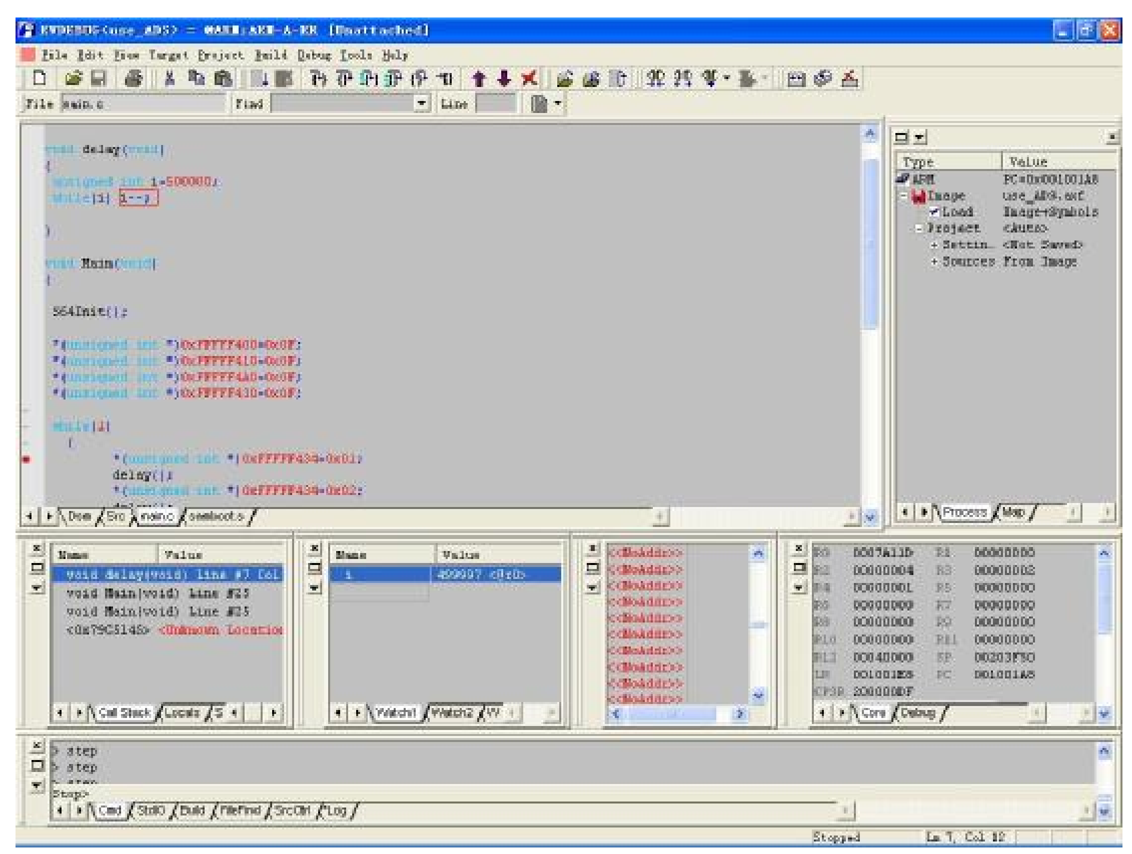
点击 Target **à** Load Image…，或者直接在下图中点击，即可加载欲调试的 image 文件。



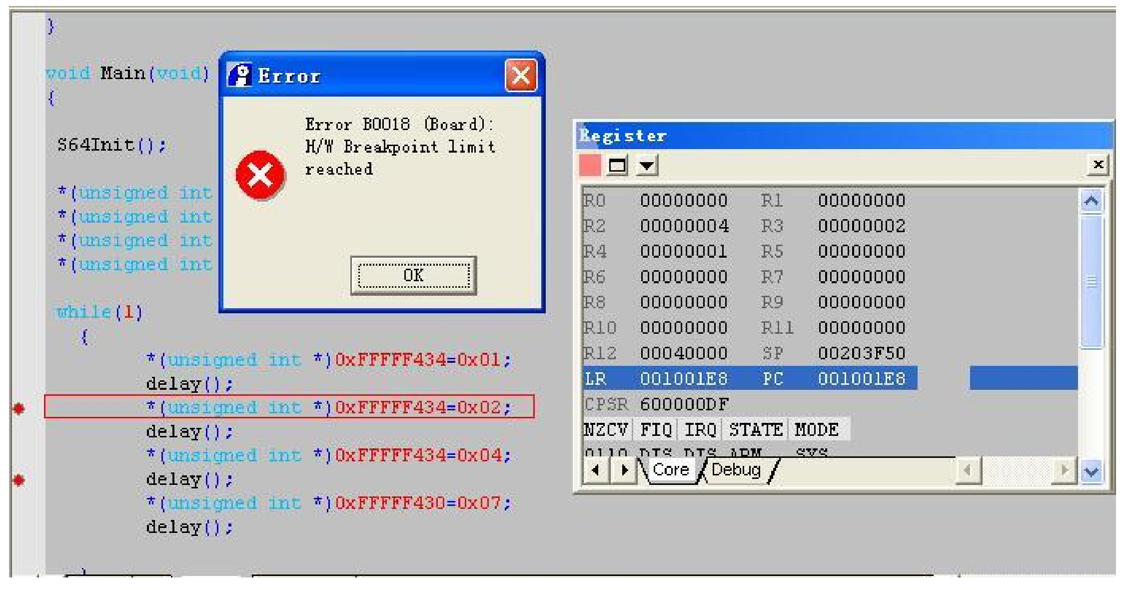
加载调试文件后 RVD 的窗口如下：



可以在源代码中设置断点并运行。可以查看变量，内存，寄存器值。



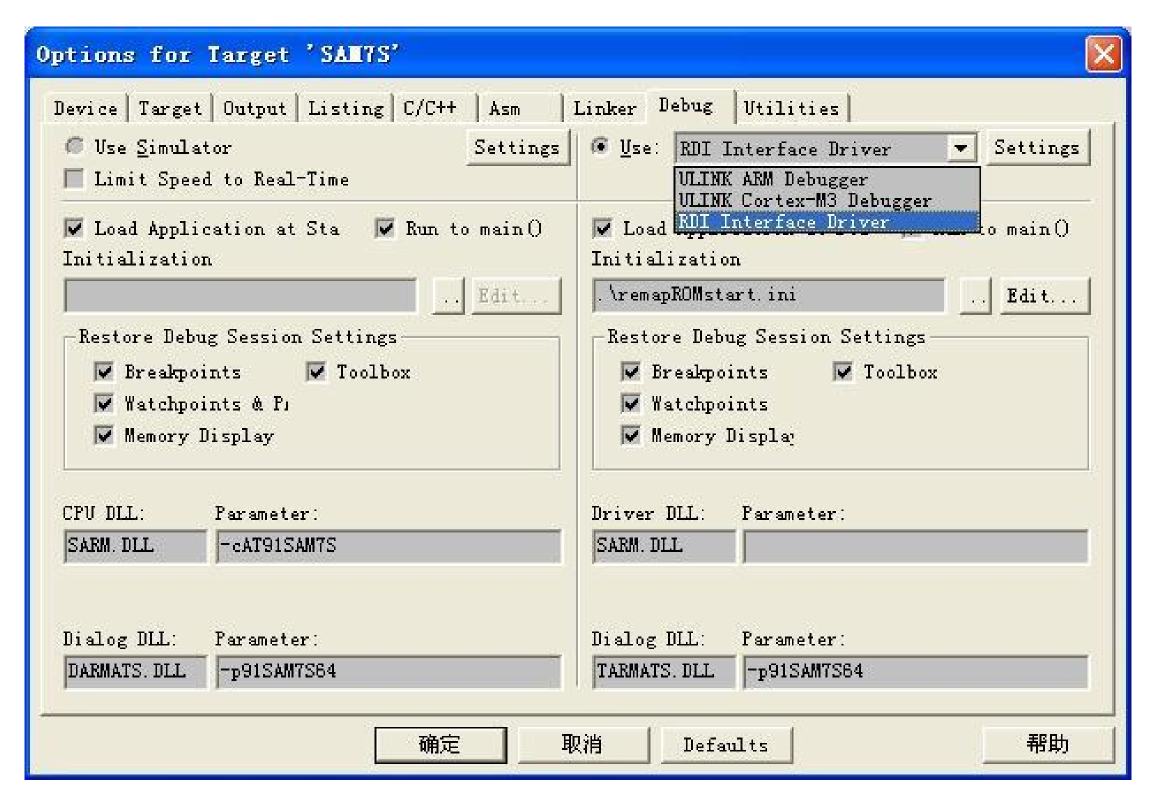
如果在设置断点的时候出现:



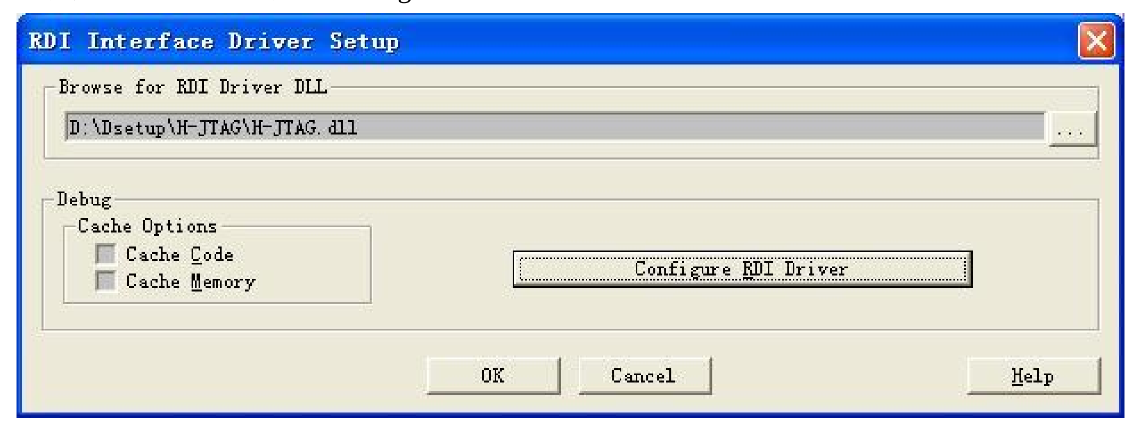
即说明硬件断点数量已用尽，应清除一些不用的断点后再试。

2.3 在 Keil 中使用 H-JTAG 调试

在 keil 中配置调试器使用 RDI 接口，如下图：

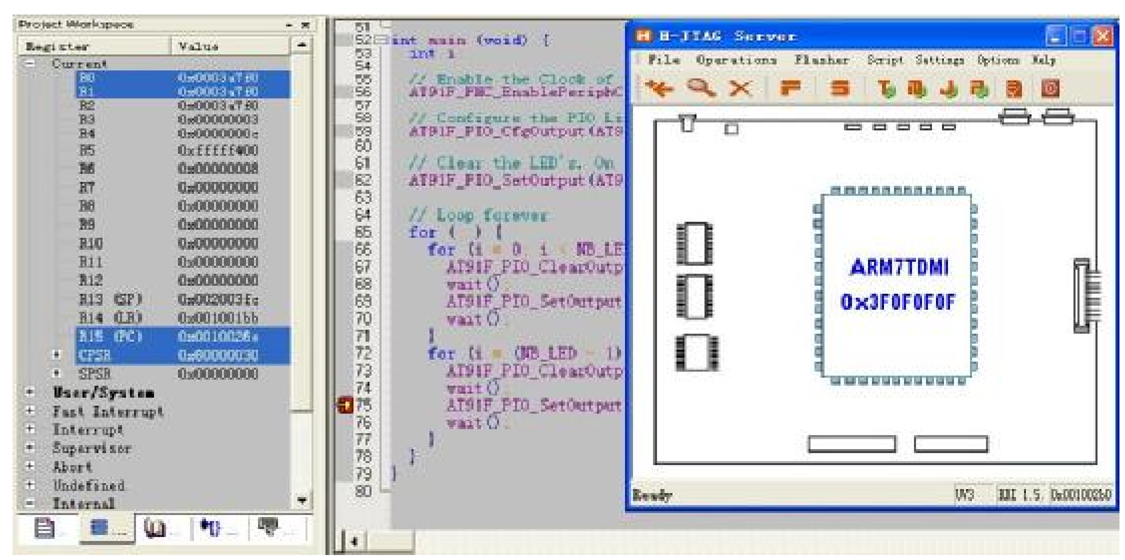


选择完成后，点击右侧的 Settings，



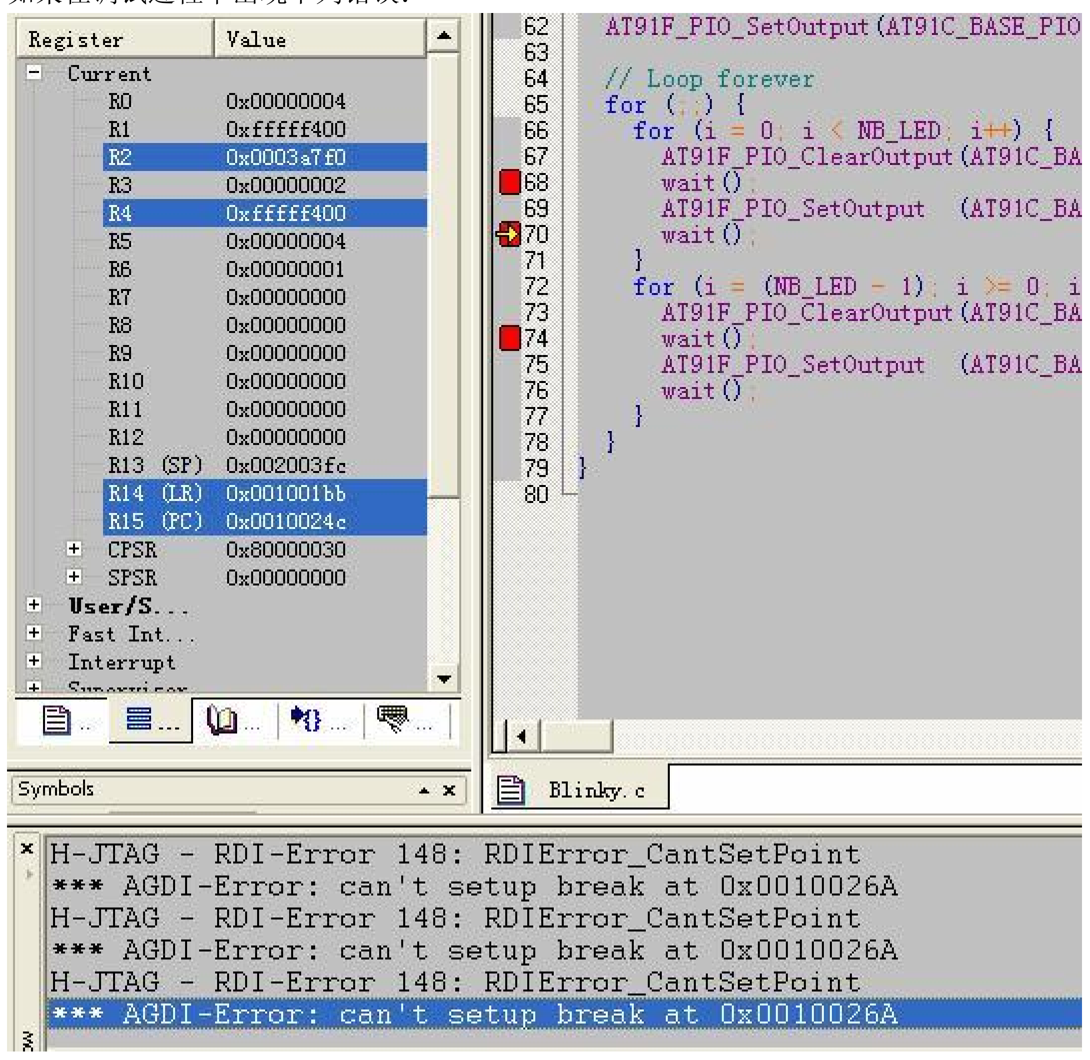
选择并添加 H-JTAG.dll，点击 OK。

点击 debug，keil 即会与 H-JTAG 建立连接，并开始调试。同时 H-JTAG 下方的状态栏也会显示连接的状态，如下图：



此时，即可开始调试。

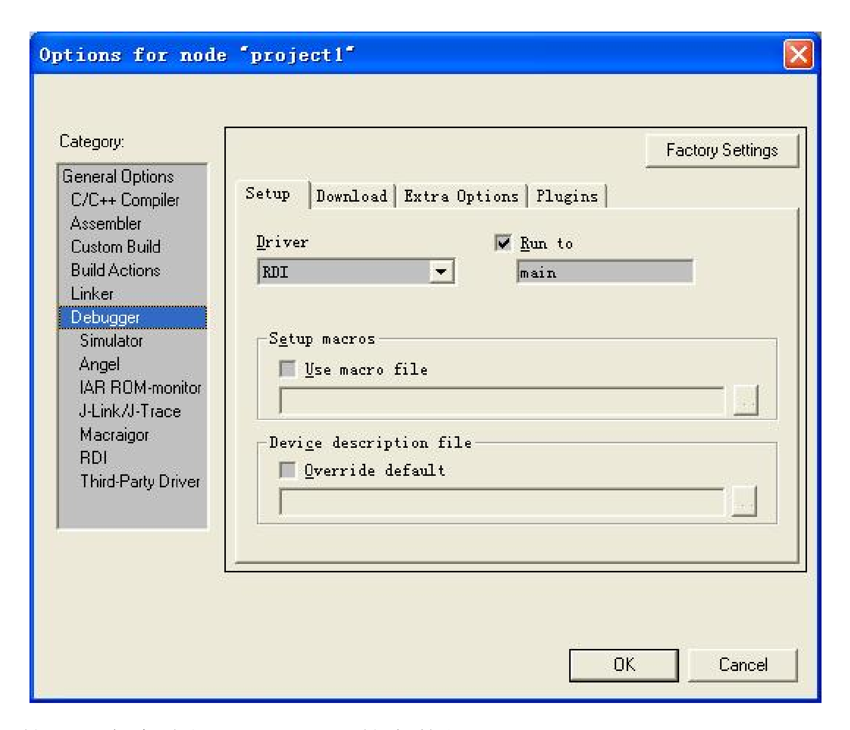
如果在调试过程中出现下列错误：



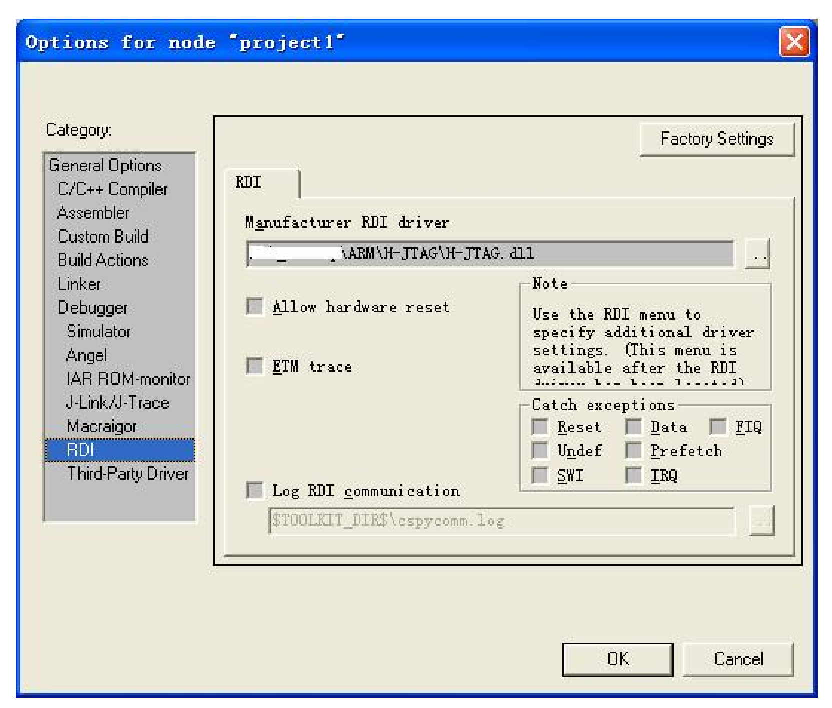
则说明断点设置过多。

2.4 在 IAR 中使用 H-JTAG 调试

在工程设置中选择 Debugger 配置页面，Driver 选择 RDI：



然后在 RDI 的配置卡中选择 H-JTAG.dll 的安装位置：



以上过程即可完成配置。以下就是调试的步骤，这里不在多说。注意在调试前运行 h－jtag，并正确找到硬件即可。

第 3 章 编程

H-JTAG 的另一个有用的功能就是 flash 编程。这对于片上 ram 较小需要在 flash 上调试的 ARM 控制器和有外部 flash 的 ARM 处理器来说，提供一种廉价但却有效的代码下载方式。

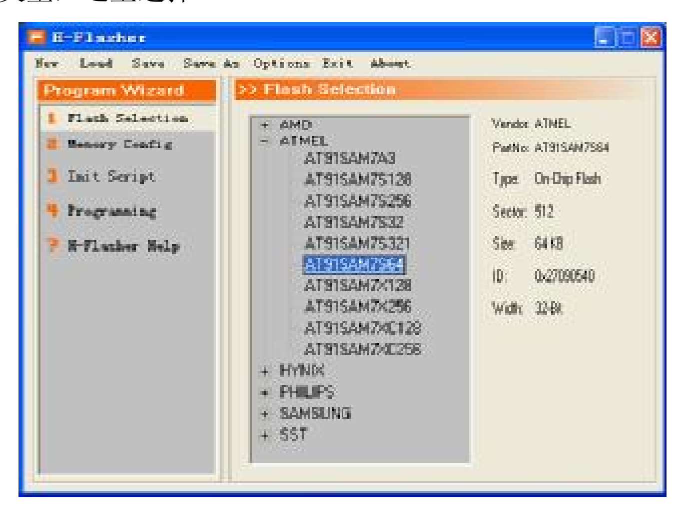
3．1 AT91SAM7S64

下面以 MCUZONE 的 S64－DEK 为例说明如何编程 flash。

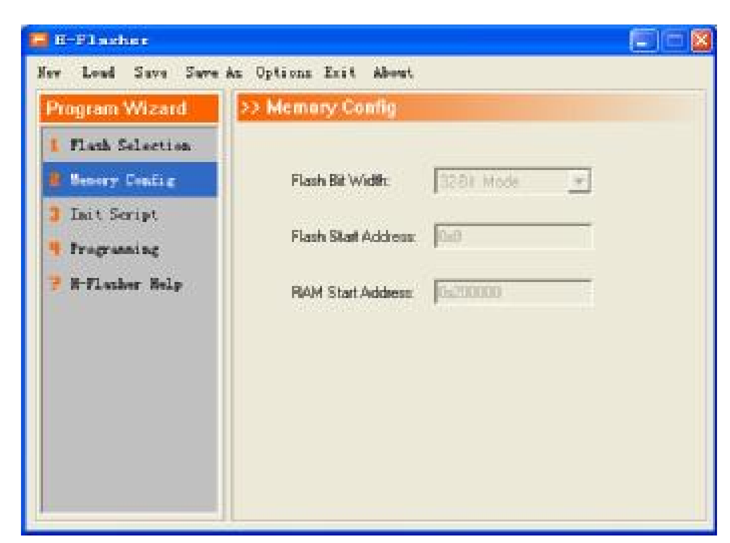
首先连接好板子的 USB 线，这样就会给板子上电。然后连接好 wiggler 与开发板。运行 h－jtag，将会发现 ARM7 内核，如果不能找到，请检查连接。找到内核后，运行 h－flasher，

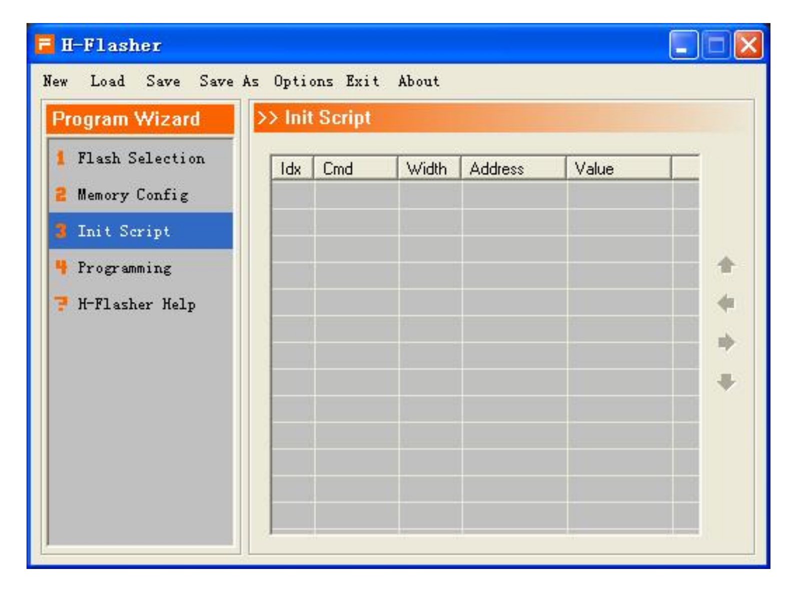


选择正确的 flash 类型，这里选择 AT91SAM7S64。

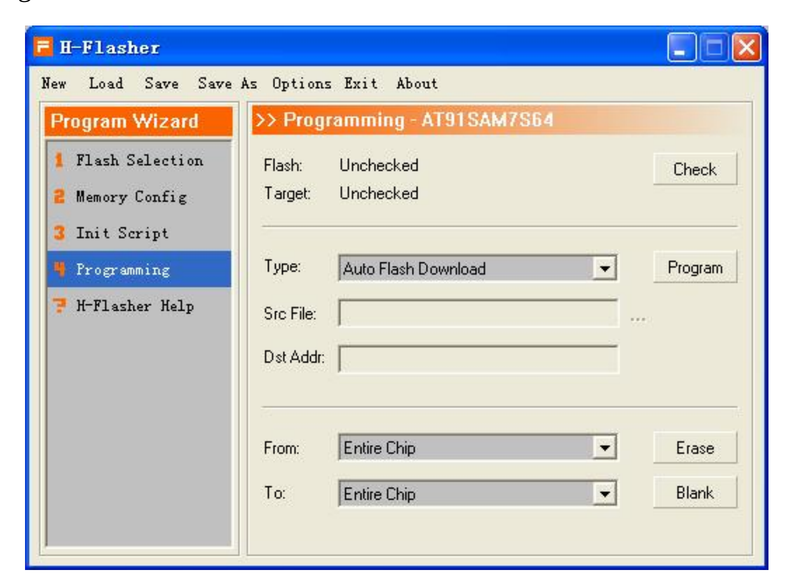


由于这是内部整合的 flash，所以不需要过多的设置， Memory Config 和 Init Script 可以不用修改：

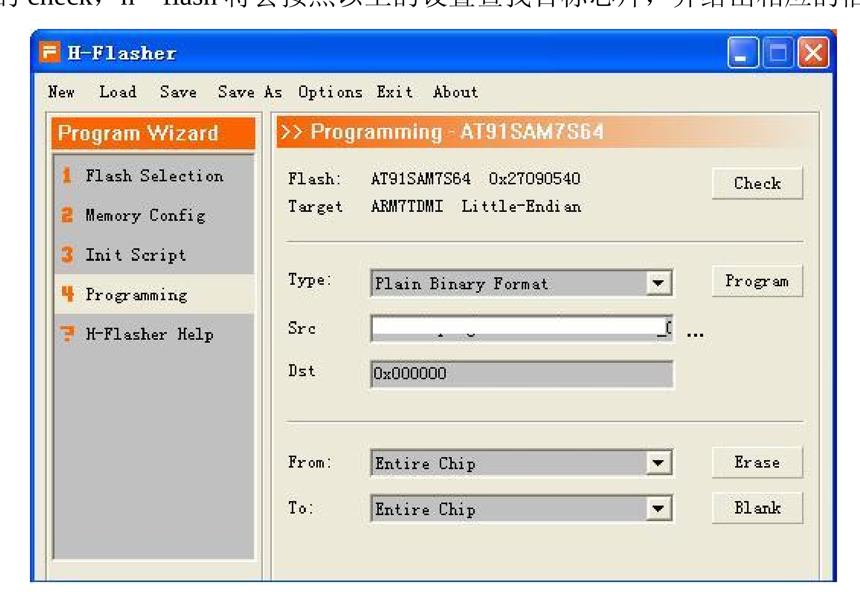




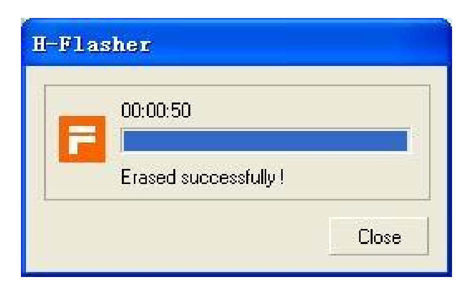
在 Programming 页面进行实际的编程操作。



点击左边的 check，h－flash 将会按照以上的设置查找目标芯片，并给出相应的信息：



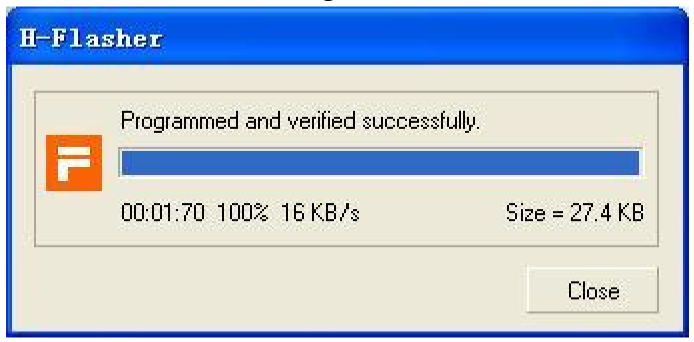
以上就是正确连接上后，找到信息。此时点击下方的 Erase，擦除芯片：



点击 Blank 可以检查 flash 是否擦除完成。

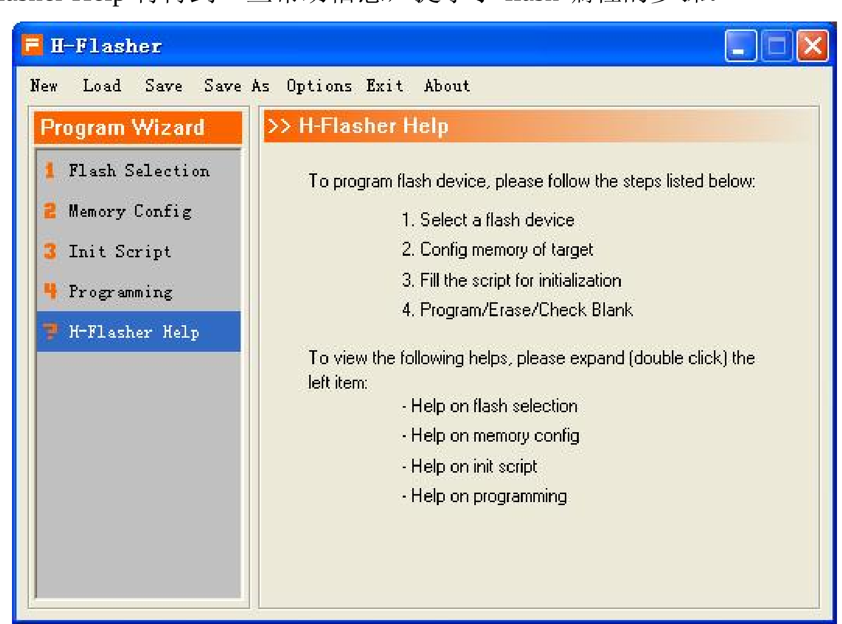


在中间的 Program 选项卡中选择目标文件的类型，再设置好目标文件的路径，下方的 Dst 设置为 0，也就 flash 的地址。然后点击 Program：



以上就是编程完成的画面。

选择 H－Flasher Help 将得到一些帮助信息，提示了 flash 编程的步骤。



上方菜单栏中的主要功能在于加载和保存 flash 编程的设置，方便多次使用同一设置。

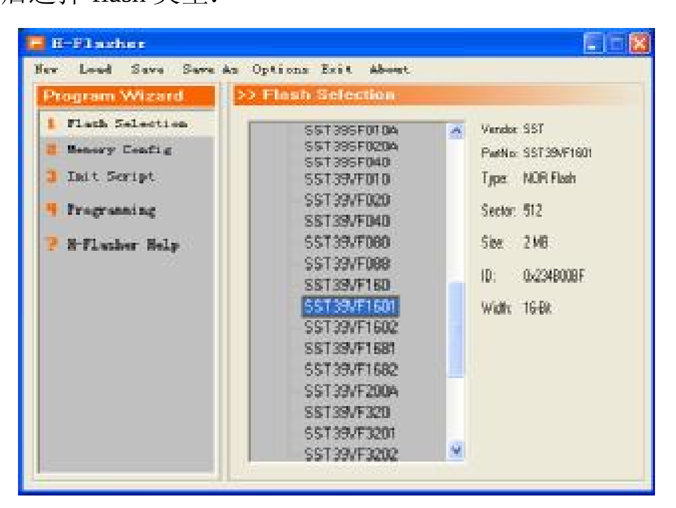
3．2　S3C44B0 公版

44B0 的公版使用了外置的 flash，大都是 SST 的产品。下面以一块装配了 SST39VF1601 的板子为例说明编程方法。

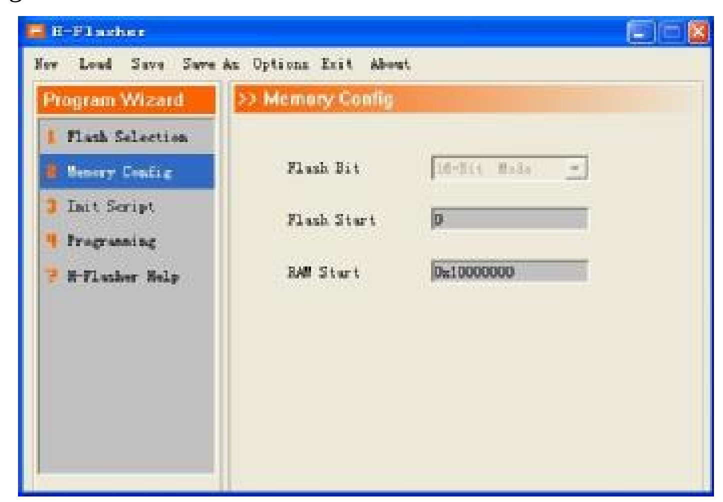
正确连接硬件后运行 h－jtag：



运行 h－flasher 后选择 flash 类型：

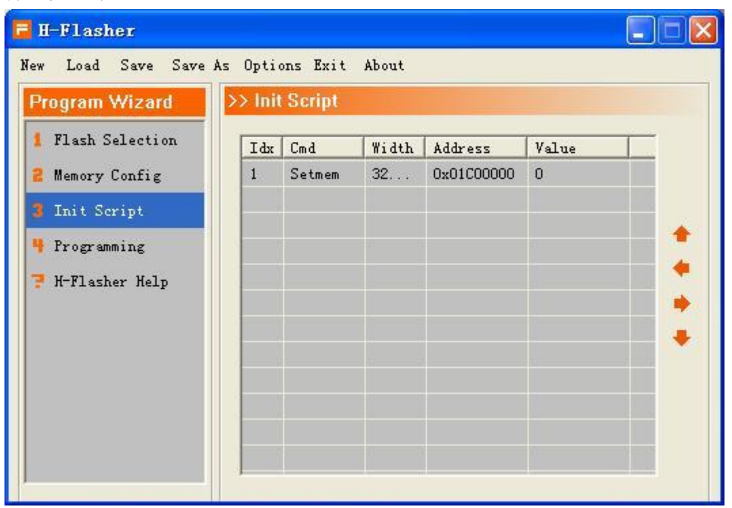


Memory Config 中设置如下：



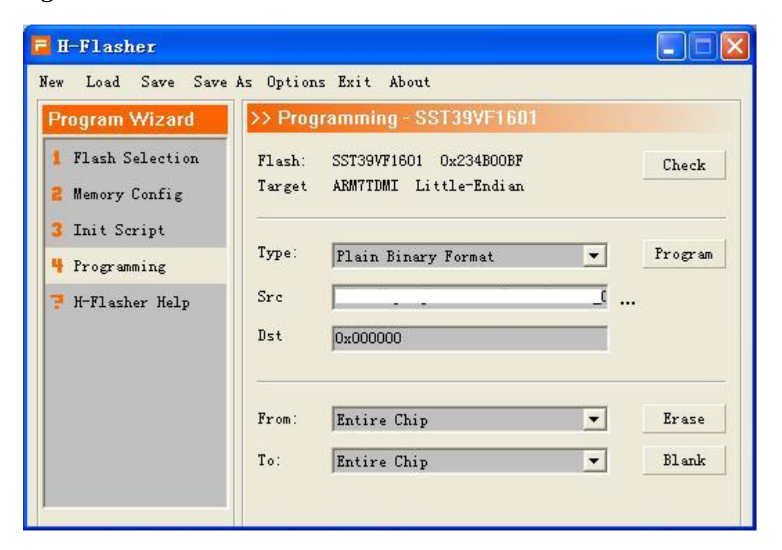
Flash 的起始地址为 0（根据 44B0 公版定义），RAM 起始地址设置为 S3C44B0 内部 RAM 的起始地址。

Init Script 中需要设置如下：



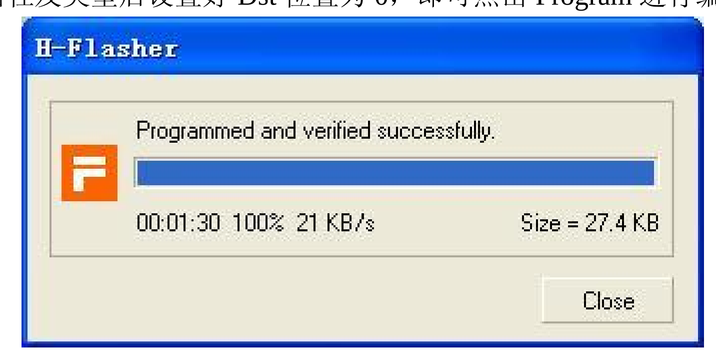
需要向 0x01C00000 处写入 0（32 位写操作），禁止 WDT。

在 Programming 选项卡中点击 check：



可以看到读出的 flash 芯片的 ID 与处理器核心类型。

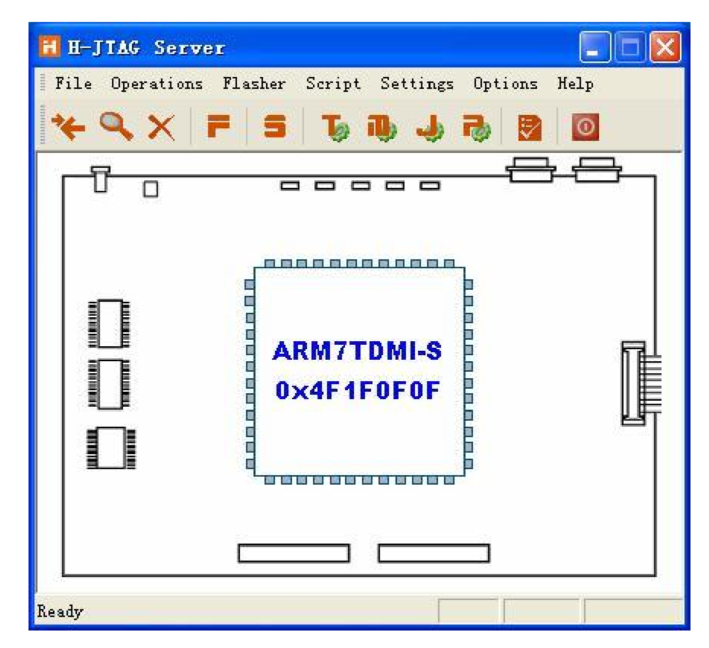
选择好文件路径及类型后设置好 Dst 位置为 0，即可点击 Program 进行编程：



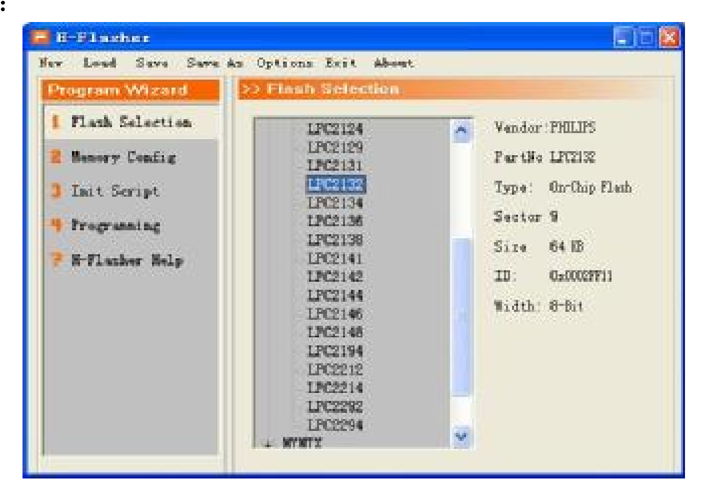
3．3 LPC2132 测试板

由于 LPC 使用了 ARM7TDMI-S 内核，注意 LPC 的 RTCK 需要下拉才能进入 JTAG 模式，以下给出编程的步骤。

正确连接硬件后运行 h－jtag：



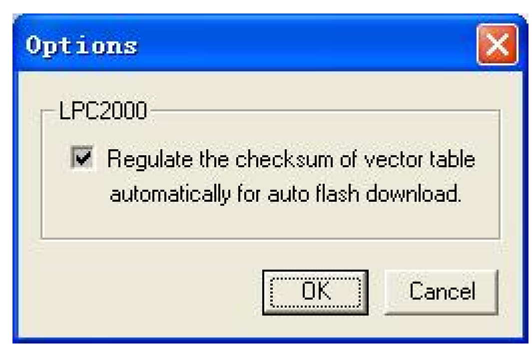
选择器件类型：



器件检测：



以下的编程步骤与其它相同。对于 LPC2000 系列有个特殊的选项。点击菜单中的 Optinos：



勾选这个选项将在编程时自动生成 LPC2000 系列所需要的向量之和。

附录 A　Wiggler 电路图

